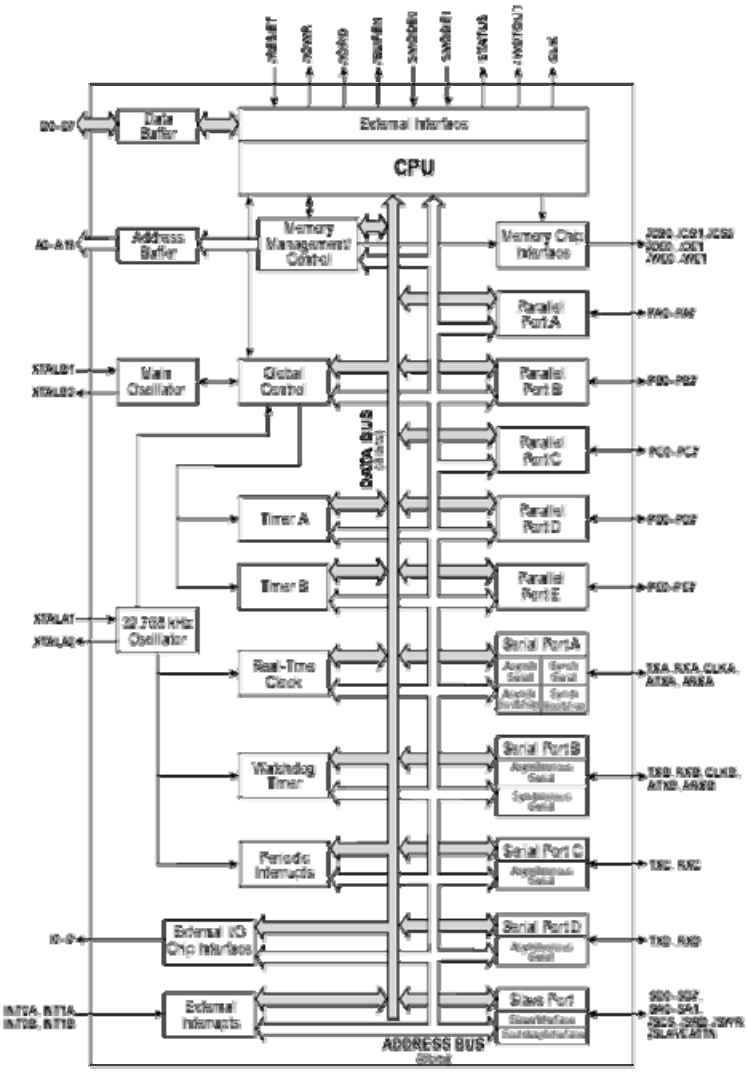
แสดงข้อความบนจอ LCD ผ่าน RS232 (RCM2200)

โดย : นายสมเกียรติ แข็งการ วศ.บ.คอมพิวเตอร์ มทร. (นักศึกษาฝึกงาน)
ติดต่อ : smokeec16@hotmail.com

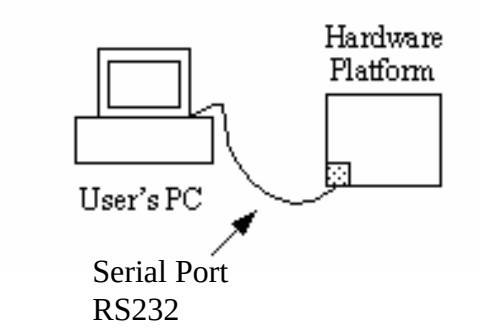
อีกหนึ่งความสามารถของชุดพัฒนา RCM2200 ที่มีจอ LCD และ Port RS232 ให้ใช้งาน ซึ่งในโครงงานนี้ได้นำชุดพัฒนา RCM2200 เชื่อมต่อกับ PC ผ่านทาง RS232 โดยใช้โปรแกรม HyperTerminal ในการพิมพ์ข้อออกทางจอ LCD

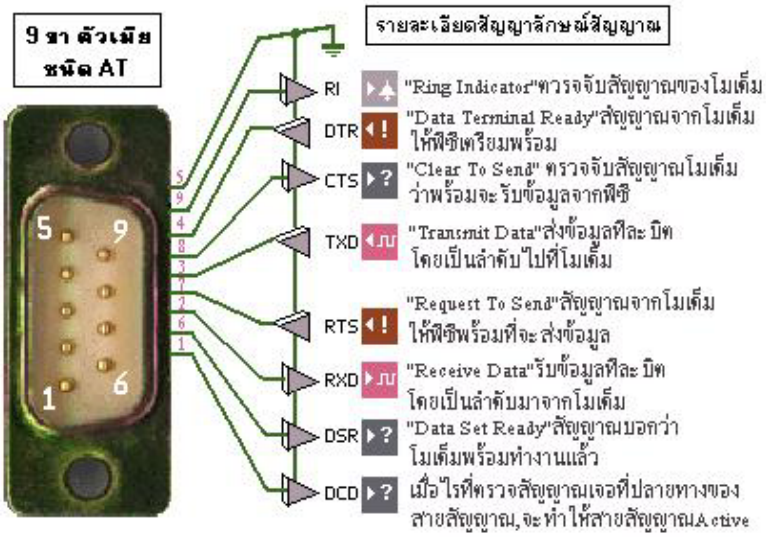
จอ LCD ที่ใช้งานขนาด 5x8 จุด 16 ตัวอักษร 2 บรรทัด ต่อเข้ากับ Parallel Port A เป็นขา Data ทั้งหมด 4 ขาทำงานในโหมด 4 Bit และต่อเข้ากับ Parallel Port E เป็นขา Control 3 ขา และ Serial Port ผ่าน RS232 เพื่อรับข้อมูลจาก PC

Block diagram of Rabbit



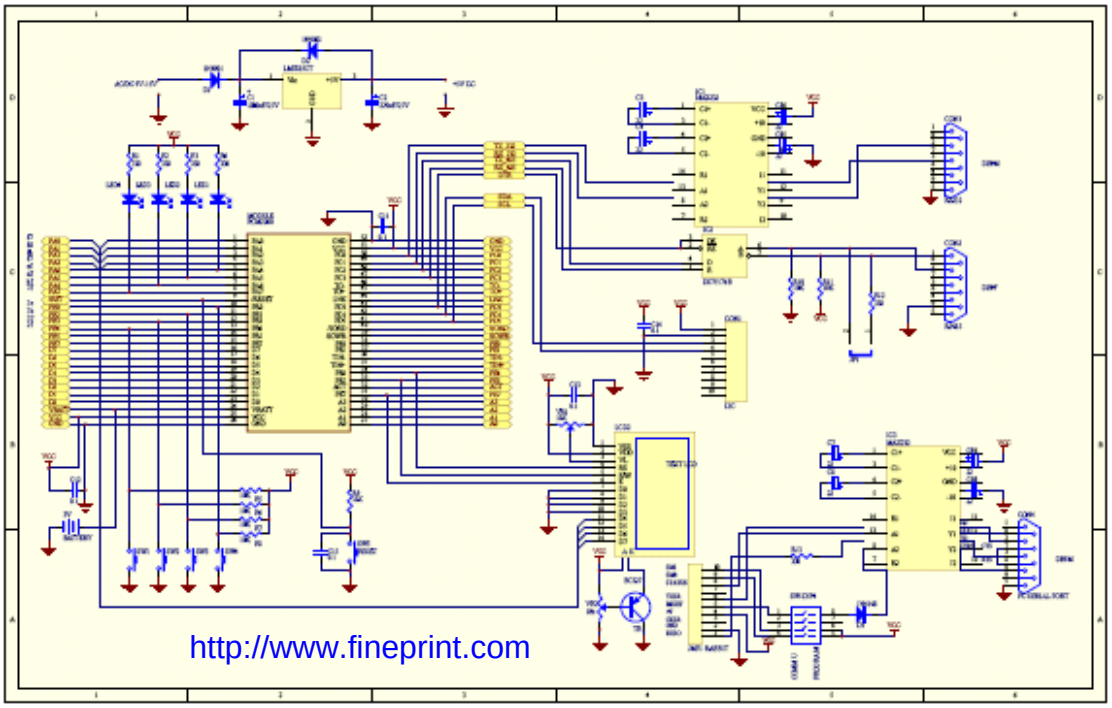
การเชื่อมต่อ





การเชื่อมต่อใช้มาตรฐาน RS232 เชื่อมต่อเข้ากับ COM Port ของ PC โดยใช้โปรแกรม HyperTerminal ในการติดต่อสื่อสาร ส่วนปลายอีกด้านหนึ่งต่อเข้ากับ Serial Port ของชุดพัฒนา RCM2200 โดยสัญญาณ TX ของด้านหนึ่ง จะต้องต่อเข้ากับสัญญาณ RX ของอีกด้านหนึ่ง

Schematic



การโปรแกรม RCM2200

โปรแกรมที่ใช้พัฒนาคือโปรแกรม Dynamic C ของ Z-WORLD และยังมี Library และโปรแกรมตัวอย่างต่าง ๆให้ผู้ใช้ด้วย

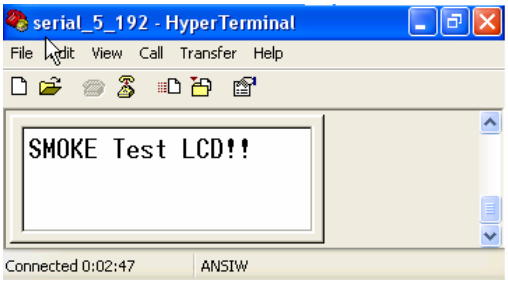
เริ่มต้นเขียนโปรแกรม

เลือกโปรแกรมโดยไปที่ Start > All Programs > Dynamic C Premier 7.25P > Dynamic C 7.25P > File > New เพื่อสร้างงานขึ้นใหม่ จากนั้นพิมพ์ Code โปรแกรมตามตัวอย่างข้างล่างนี้ เสร็จแล้วก็ Save และ Compile โดยการกดปุ่ม F5 จากนั้นเริ่มต้นการทดสอบโปรแกรม

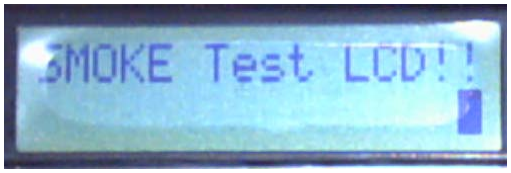
การทดสอบโปรแกรม

เปิดโปรแกรม HyperTerminal ขึ้นมา สร้าง Connection ขึ้นมาใหม่โดยตั้งค่าต่าง ๆดังนี้ 19200, 8, None, 1 จากนั้นทำการพิมพ์ข้อความที่โปรแกรม HyperTerminal ข้อความจะปรากฏที่โปรแกรม HyperTerminal และที่จอ LCD ด้วย

ส่วนของโปรแกรม HyperTerminal



ส่วนของจอ LCD



Source code

```
#define LCD_RS 4                                    //Set RS is Port E 4
#define LCD_RW 5                                    //Set RW is Port E 5
#define LCD_E 7                                     //Set Enabled is Port E 7
#define DINBUFSIZE 15                               //Set serial port buffer
#define DOUTBUFSIZE 15

char serX[256];
const char str1[] = {" SMOKE TEST LCD "};           //Text to test
const char str2[] = {" SMOKE TEST LINE"};           //Text to test
const static char addline[2] = {0x80, 0xc0};        //Command for write text to line 1 and line 2
unsigned char LCD_ADDR, LCD_LINE;

/*      Delay Function      */
void Delay(int iDelay){
        unsigned long ul0;
        ul0 = MS_TIMER;                             // get current timer value
        while ( MS_TIMER < ul0 + (unsigned long) iDelay );
}

/*      Initial LCD         */
void Lcd_int(void)  {
        WrPortI(SPCR, &SPCRShadow, 0x84);           //Set Port A to Output
        WrPortI(PADR, &PADRShadow, 0xFF);           //Reset Port A
        WrPortI(PEDDR,&PEDDRShadow,0xb0);           //Set Port E to Output
        BitWrPortI(PEDR,&PEDRShadow,0,LCD_RW);      //LCD_RW = 0

        BitWrPortI(PEDR,&PEDRShadow,0,LCD_E);       //LCD_E = 0
        BitWrPortI(PEDR,&PEDRShadow,0,LCD_RS);      //LCD_RS = 0
        Delay(50);                                  //Delay 50 mSec

        Write_Command(0x33);                        //Set to 4 Bit Mode
        Write_Command(0x33);
        Write_Command(0x33);
```

```
        Write_Command(0x22);

        Write_Data(0x28);                                   //Set 4 Bit Mode 5x7 Pixel
        Write_Data(0x01);                                   //Clear Display
        Write_Data(0x10);                                   //Move curser to left
        Write_Data(0x06);                                   //Display string and move curser to right
        Write_Data(0x0c);                                   //Display on and curser off

        BitWrPortI(PEDR,&PEDRShadow,1,LCD_RS);              //LCD_RS = 1
        serDopen(19200);                                    //Set  Bit per Sec
}

Write_Command(char Cmdcode) {
        BitWrPortI(PEDR,&PEDRShadow,1,LCD_E);               //LCD_E = 1
        WrPortI(PADR,&PADRShadow,Cmdcode&0x0F);             //Write command to Port A
        BitWrPortI(PEDR,&PEDRShadow,0,LCD_E);               //LCD_E = 0
        Delay(3);                                           //Delay 3 mSec
        return 0;
}

Write_Data(char Dispdata)  {
        BitWrPortI(PEDR,&PEDRShadow,1,LCD_E);               //LCD_E = 1
        WrPortI(PADR,&PADRShadow,Dispdata>>4);              //Write Data to Port A
        BitWrPortI(PEDR,&PEDRShadow,0,LCD_E);               //LCD_E = 0

        BitWrPortI(PEDR,&PEDRShadow,1,LCD_E);               //LCD_E = 1
        WrPortI(PADR,&PADRShadow,Dispdata&0x0f);            //Write Data to Port A
        BitWrPortI(PEDR,&PEDRShadow,0,LCD_E);               //LCD_E = 0
        Delay(3);                                           //Delay 3 mSec
        return 0;
}

void Line(char wline){
        BitWrPortI(PEDR,&PEDRShadow,0,LCD_RS);              //LCD_RS = 0
        LCD_ADDR = addline[wline-1];
        Write_Data(LCD_ADDR);
        BitWrPortI(PEDR,&PEDRShadow,1,LCD_RS);              //LCD_RS = 1
}

/*      Write String Function       */
Write_str(unsigned char *str){
        while(*str != 0)
                Write_char(*str++);
        return 0;
}

/*      Write Charector Function  */
Write_char(unsigned char chr){
        BitWrPortI(PEDR,&PEDRShadow,1,LCD_RS);              //LCD_RS = 1
        Write_Data(chr);                                    //Write Data
        LCD_ADDR++;
        if(LCD_ADDR == 0x90) Line(2);                       //Goto Line 2
        if(LCD_ADDR == 0xD0) Line(1);                       //Goto Line 1
        return 0;
}

/*      Main Function     */
void main(void){
        int i,c;
        serX[0]=0;                                          //Clear serX
```

```
        Lcd_int();                                  //Initializing LCD
        Line(1);                                    //Start on Line 1

        while(1){
                if((c = serDgetc()) != -1){
                        sprintf(serX,"%c",c);       //Get charector to serX
                        serDputc(c);                //Put charector to Hyperterminal
                        Write_str(serX);            //Write string to LCD
                }
        }

        serDclose();                                //Close serial Port
}
```

เอกสารอ้างอิง

- JAN AXELSON, Embedded ETHERNET AND INTERNET COMPLETE, Lakeview Research LLC, 2003
- http://www.ylib.com.cn/
- http://www.rabbitsemiconductor.com
- http://www.embeddedj.co.th